NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

DYNAMICS OF ABUSIVE IPv6 NETWORKS

by

Mark J. Turner

September 2014

Thesis Advisor:	Robert Beverly
Second Reader:	Casey Deccio

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	22–09–2014	Master's Thesis 03-25-2013 to 09-26-2014

4. TITLE AND SUBTITLE

DYNAMICS OF ABUSIVE IPv6 NETWORKS

5. FUNDING NUMBERS

CNS-1111445

6. AUTHOR(S)

Mark J. Turner

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Science Foundation
4201 Wilson Blvd., Arlington, VA 22230

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

The exhaustion of available public IPv4 addresses has had a significant impact in an expanding, networked world and has led to increased adoption of IPv6. As IPv6 becomes more commonplace, it permits abusive and malicious parties to exploit both new and existing vulnerabilities. Among such vulnerabilities is abusive electronic messaging, or spam. To better understand the impact of spam utilizing IPv6 as its delivery protocol, this study focused on both real-world IPv6 spam collected from large production domain and IPv6 spam laboratory measurements. This study used various network traffic analysis tools to detect, classify, and associate IPv6 spamming behavior, both at the victim mail exchanger and among IPv6 wide-area routes. Furthermore, popular mail transfer agents were tested in an effort to profile their IPv6 behavior and correlate with spam obtained from the real world production domain. Results show that while IPv6 spamming behavior is growing, it is still in its infancy and no outstanding characteristics emerged that allow for definitive classification as a dominant IPv6 spamming behavior.

14. SUBJECT TERMS	15. NUMBER OF PAGES
IPv6, Spam, SMTP, BGP, MTA, Security	65
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

DYNAMICS OF ABUSIVE IPv6 NETWORKS

Mark J. Turner
Lieutenant, United States Navy
B.S., United States Naval Academy, 2006

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2014

Author: Mark J. Turner

Approved by: Robert Beverly
Thesis Advisor

Casey Deccio
Second Reader

Peter J. Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The exhaustion of available public IPv4 addresses has had a significant impact in an expanding, networked world and has led to increased adoption of IPv6. As IPv6 becomes more commonplace, it permits abusive and malicious parties to exploit both new and existing vulnerabilities. Among such vulnerabilities is abusive electronic messaging, or spam. To better understand the impact of spam utilizing IPv6 as its delivery protocol, this study focused on both real-world IPv6 spam collected from large production domain and IPv6 spam laboratory measurements. This study used various network traffic analysis tools to detect, classify, and associate IPv6 spamming behavior, both at the victim mail exchanger and among IPv6 wide-area routes. Furthermore, popular mail transfer agents were tested in an effort to profile their IPv6 behavior and correlate with spam obtained from the real world production domain. Results show that while IPv6 spamming behavior is growing, it is still in its infancy and no outstanding characteristics emerged that allow for definitive classification as a dominant IPv6 spamming behavior.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1 Introduction **1**
1.1 Motivation 2
1.2 Research Questions 4
1.3 Contributions 4
1.4 Thesis Structure 5

2 Technical Review **7**
2.1 Malicious Traffic 7
2.2 BGP Routing 8
2.3 Related IPv6 Malicious Traffic Studies 10

3 Methodology **13**
3.1 Experimental Production Domain 14
3.2 Laboratory Spamming Testbed 16
3.3 BGP Correlation Algorithm 20

4 Experimental Results **23**
4.1 Notable `example.com` Analysis 23
4.2 MTAs Behaving Badly 25
4.3 Tracking IPv6 Spam via BGP 30

5 Conclusion **37**
5.1 Future Work 37

Appendix **39**

List of References **43**

Initial Distribution List **47**

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

Figure 1.1 Google User IPv6 Adoption Statistics 3

Figure 2.1 Example Network Displaying IPv6 Network Complexity 9

Figure 3.1 `example.com` Experimental Setup 15

Figure 3.2 `test.com`'s Operational Behavior 18

Figure 3.3 Example BGP Route Hijacking 20

Figure 3.4 BGP Correlation Algorithm Pseudocode 22

Figure 4.1 Weekly `connect` Attempts to `example.com`'s Sensor MTA. . . . 25

Figure 4.2 Length of Short-Lived BGP Episodes Lasting Under 25 Hours . . 31

Figure 4.3 Observation of `2a01:111::/32` BGP Agility Behavior 32

Figure 4.4 `2a01:111::/32` BGP Agility Episodes Lasting Less Than 25 Hours 32

Figure 4.5 Observation of `2607:9000::/32` BGP Agility Behavior 34

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

Table 3.1 DNS Resource Records Corresponding to `test.com`. 17

Table 3.2 Test Domain Configuration 18

Table 4.1 Summary of Results from IPv6/IPv4 Address Association. 24

Table 4.2 `example.com`'s Inferred Operating Systems for Spamming Hosts . 25

Table 4.3 Results from `test.com` Spamming Experiment 26

Table 4.4 Statistics from `example.com` Dataset 30

Table 4.5 BGP Algorithm Statistics from BGP Episodes Under 25 Hours . . 30

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

A	A record
AAAA	quad-A record
AS	autonomous system
ASN	Autonomous System Number
BGP	Border Gateway Protocol
DNS	Domain Name System
DNSBL	Domain Name System blacklist
DoS	denial of service attack
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
ICMPv6	Internet Control Message Protocol version 6
IDS	intrusion detection system
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISP	Internet Service Provider
MRT	multithreaded routing toolkit
MTA	Mail Transfer Agent
MX	Mail Exchanger
MXS	Microsoft Exchange Server 2010

NAT	Network Address Translation
OS	Operating System
pcap	packet capture file
p0f	passive operating system fingerprint
RDNS	Reverse Domain Name System
RFC	Request For Comments
RIB	routing information base
RIR	Regional Internet Registries
RouteViews	University of Oregon RouteViews Project
SMTP	Simple Mail Transfer Protocol
TCP	Transmission Control Protocol

Acknowledgments

I want to thank Dr. Robert Beverly for helping me navigate both this thesis process and the Computer Science Department at NPS. From the beginning, this project seemed daunting, but he always found time to share his expertise and guide my research and writing. His quest for knowledge and desire to explore the details of network measurement inspired me and transformed this research from a required project to an exhilarating experience.

I would also like to thank Dr. Casey Deccio for his work developing the direction of research. His detailed nature and thirst for spam classification really pushed the difficult points. He was always available for questions during this research, and I thank him for his guidance and expertise.

Finally, I would like to thank my wife, Jennifer, for her unwavering support throughout this process. My studies at NPS were valuable and filled with time-consuming requirements. I could not have done it without her. She can finally update her status, knowing "School Deployment" is over, and she will no longer be a "Computer Science Widow."

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

An Internet Protocol (IP) address is a distinct 32 bit or 128 bit unsigned integer that is assigned to network interfaces connected to the Internet. The Internet Assigned Numbers Authority (IANA) is responsible for assigning IP address blocks to regional Regional Internet Registries (RIR), which, in turn, assign those blocks to conglomerates and individuals based on geographic localities. An IP address is either an Internet Protocol version 4 (IPv4) address (32-bit) or an Internet Protocol version 6 (IPv6) address (128 bit) and is used to route data between devices connected to the Internet. There are about 4 billion possible IP addresses within the 32-bit IPv4 address space, but because contiguous addressing is required for routing requirements, the actable number of usable IPv4 addresses for end-hosts is much smaller. In 2011, IANA's available IPv4 address pool officially depleted [1], [2] and, as of this writing, some RIRs have only a small number of available address blocks remaining. With this exhaustion of IPv4 addresses, more and more organizations and individuals are adopting IPv6 [3].

IPv6 was created in 1998 in anticipation of the exhaustion of IPv4 addresses, but it was not initially widely adopted due to an increased complexity in addressing, cost associated with equipment upgrades, management cost, and the availability of interim solutions such as Network Address Translation (NAT) [4]. IPv6 is formally defined in Request For Comments (RFC) 2460. IPv6 is not specifically interoperable with IPv4, and essentially acts as a parallel, independent network of IPv4 [5]. To facilitate network traffic between the differing protocols, an IPv6 network can use a translation technology such as 6to4, 6in4, or Teredo tunneling protocol that allows IPv6 traffic to be encapsulated and routed over IPv4 networks. Today, major network service providers, equipment manufacturers, and the U.S. government utilize IPv6 in both their operating systems and network switching devices in order to take advantage of IPv6 and its addressing capability [6], [7].

Due to the rise in IPv6 adoption, it is natural to expect a corresponding rise in malicious traffic that is routed using IPv6. Using IPv6 has the potential to afford attackers and malicious traffic several advantages. Since IPv6 has less usage in a production environment than

IPv4, firewall configurations detecting malicious IPv6 traffic are not as well documented, configured, and deployed as their IPv4 counterparts. intrusion detection systems (IDSs) rely on expected traffic profiles to catch anomalous or malicious behaviors, most of which use profiles that have little to no exposure as to how a normal IPv6 traffic profile presents. Another vulnerability in IPv6 that does not exist in IPv4 lies in how Internet Control Message Protocol version 6 (ICMPv6) is used. Firewalls must permit ICMPv6 traffic in order to allow IPv6 to operate correctly. IPv4 firewalls traditionally block Internet Control Message Protocol (ICMP) traffic due to widely known vulnerabilities, but ICMPv6 messages must be allowed through firewalls for IPv6 services to function correctly, even though without careful configuration ICMPv6 messages can be easily used to conduct denial of service attacks (DoSs) or profile networks [8]. Since some systems use IPv4-IPv6 tunneling technologies, it does not take a great deal of effort for a malicious entity to inject malicious traffic if they know which routers are being used to tunnel IPv6 traffic over an IPv4 network. Until such a time that IPv6 is utilized and studied as much as IPv4, more and more vulnerabilities will be discovered and exploited in IPv6, making it more advantageous for a nefarious user to use IPv6 in the early stages of its deployment [9].

The goal of this thesis is to have a better understanding of how one form of malicious and abusive traffic, spam, flows through the Internet with IPv6. Due to this anticipated rise in abusive IPv6 traffic, we endeavored to further the study of spam in IPv6 by examining collected spam traffic from a public facing mail server, creating a laboratory testbed to mimic IPv6 spamming behavior, and correlating spam data to a Border Gateway Protocol (BGP) dataset to analyze spamming behavior in IPv6. This study seeks to determine any classifiable behaviors that can aid malicious traffic mitigation techniques. Ultimately, new metrics could be used to classify spam over IPv6 based on traffic characteristics alone, resulting in less complex filtering at endpoint Mail Exchangers (MXs), a denial of spam routing between autonomous systems (ASs), and a more concrete understanding of malicious traffic techniques in IPv6 [10].

1.1 Motivation

The adoption of IPv6 is experiencing an increasing trend, 10 fold since 2008, which replenishes usable IP addresses for networked devices but also allows abusive and malicious parties to exploit new and existing vulnerabilities [3].

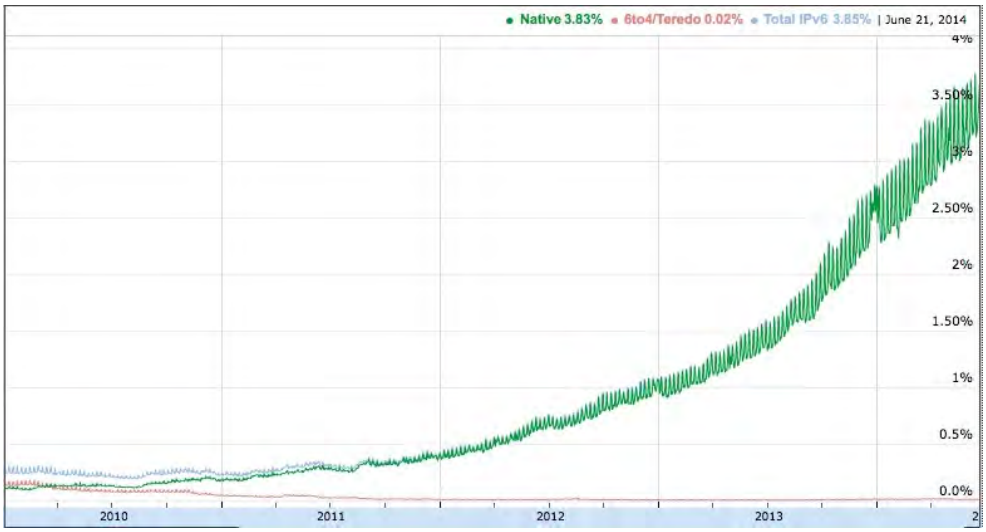

Figure 1.1: Google User IPv6 Adoption Statistics, from June 2014 [3]

For example, as a result of expanding use of IPv6, it is hypothesized that a significant amount of spam will eventually utilize IPv6 as its delivery protocol in order to take advantage of the exploitability of an increased address space. Such an increased address space affords spammers the ability to use a different source address for each spam message sent, which could allow the spammer to evade spam filters based on IP addressing and reputational data. Additionally, the IPv6 address space is so incredibly large that it is highly likely that a single device has multiple IPv6 addresses. This capability will allow an attacker with a single device to use the different interfaces to send multiple instances of malicious traffic, which would minimize the attackers footprint from a single source perspective and reduce the likelihood of malicious traffic attribution. Due to this anticipated rise in abusive IPv6 traffic, more studies need to be conducted into the behavior of spam in IPv6 [11]. While this thesis focuses on collected, real-world IPv6 spam analysis from a large production IPv6 domain, the lessons learned offer more general guidance for understanding the use and evolution of various kinds of malicious IPv6 traffic. If a concrete metric is established of spam using IPv6, numerous organizations, including the U.S. government, could benefit from a more robust method of discovering, filtering, and defeating malicious IPv6 traffic. Today, a plethora of work has been focused on spam behavior and its filtering in IPv4. Previous work on IPv6 spamming behavior has been largely theoretical and lacked any true measurement of IPv6 spamming behavior observed “in the wild.” Thus, we attempt to

observe and classify spam data and associate known spammer techniques used in IPv4 to those used in IPv6.

1.2 Research Questions

This thesis relies on collected IPv6 spam data from an enterprise level production domain. Using captured IPv6 spam packets from our production domain, experimental data collected from our laboratory spamming testbed, and BGP routing updates, we seek to characterize abusive IPv6 traffic. In doing so, we explore the following:

- Does a classifiable relationship exist between IPv6 spam and BGP routing behaviors?
- Is the larger IPv6 address space a significant resource for a spammer?
- If attackers exploited the large IPv6 address space and constantly changed their IP address would the behavior be detectable?
- Does a set of spam behaviors or metrics exist that would allow for spam to Mail Transfer Agent (MTA) correlation?
- Does a discernible set of spamming characteristics exist in IPv6 traffic that would allow an effective spam filter to be built?

1.3 Contributions

Our research efforts in malicious IPv6 traffic analysis yielded the following findings:

- IPv6 spam is two orders of magnitude less prevalent than IPv4 spam as measured in our `example.com` dataset.
- Although standards are clearly defined in various RFCs, network configurations, Operating Systems (OSs), and MTAs do not always follow most preferred network configuration preferences, making default behaviors difficult to identify.
- BGP spectrum agility is present and can be measured in IPv6 spam, but does not exactly mirror previously observed spectrum agility methods in IPv4.
- We did not discover any behaviors that would suggest that malicious IPv6 traffic is exploiting the vast IPv6 address space.

1.4 Thesis Structure

The remainder of this thesis is organized as follows:

- Chapter 2 discusses spam behavior and filtering, BGP routing, malicious traffic, and related IPv6 malicious traffic studies.
- Chapter 3 focuses on our production domain experimental setup, our laboratory spam replay domain, and introduces our efforts to correlate BGP routing behavior with IPv6 spam.
- Chapter 4 provides all the results from our laboratory spamming testbed analysis, the correlation of collected spam data to University of Oregon RouteViews Project (RouteViews) multithreaded routing toolkit (MRT) updates, and inferred IPv6 spam classification characteristics.
- Chapter 5 details thesis research conclusions and recommendations for future research areas related to this work.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Technical Review

The deployment and use of IPv6 has steadily increased since its inception in 2008, motivated in part by IPv4 address depletion, economic incentives for large-scale organizations, and promotional efforts [2], [5], [6], [12]–[14]. Events such as these have presented the possibility to understand, characterize, and protect the IPv6 Internet. There has been a significant body of work that characterizes IPv4 traffic, especially when discussing the behaviors and characteristics of spam traffic. Spam traffic is considered network traffic that can be classified as unsolicited bulk email. Oftentimes, a single source with resources will send out a mass volume of advertisement emails to incredibly large email lists that are public available for less than a few hundred dollars. A spamming bulk mailer requires a small amount of start-up cost and investment, which is well worth the effort if a spammer makes even a fraction of the sales of the products contained within the spamming messages sent each day [15]. While spam can be an excessive annoyance of poorly crafted advertising emails, they can sometimes contain attachments or links to scam offers, phishing attempts, or malicious code. Even though spam can come in many different forms, the typical motivation behind a spammer's efforts is financial gain, either through product purchasing or the compromise of personally identifiable information [16]. Unfortunately, much less attention has been applied to spam traffic on IPv6. The immense IPv6 address space and protocol differences create new vulnerabilities for exploitation and may cause a resurgence of previously solved security flaws. This chapter will review and discuss the nature of spam, how spam permeates the Internet, and previous efforts to research and classify malicious traffic.

2.1 Malicious Traffic

Malicious or abusive Internet traffic comes in many forms and has a variety of characteristics. A commonly recognized form of malicious traffic is spam. In order to properly classify malicious traffic, it is import to understand how it operates and why it is so pervasive. A spammer can send a vast volume of abusive traffic simply because spam can be easily automated, requires little management overhead, and is hard to attribute to a specific person. To focus on how spamming actually happens, one must look to how the Simple Mail Transfer

Protocol (SMTP) is exploited by a malicious actor. A legitimate email message is routed via SMTP from a sending MTA to a receiving MTA. While spam messages can be routed like legitimate email, spammers often take advantage of poorly configured or compromised networked machines, allowing the spammer to spam the receiving MTA directly without relaying through a designated MTA. By circumventing the designated MTA, spammers are also able to evade the spam filtration mechanisms deployed at the designated MTA. Spam messages can come in a variety of types, from general advertisements to specifically crafted messages at a target host machine. This wide range of malicious message variety makes it difficult to conduct message content filtering without knowing what the message is going to be before hand. As a result, SMTP abuse is mitigated IPv4 through content filters, firewall configuration, and extensive use of Domain Name System blacklists (DNSBLs), which use mail filtering lists based on known spamming hosts and IP addresses. Regrettably, analogous abusive traffic solutions for IPv6 spam are not as common nor well-tested. Since mitigation strategies in regard to IPv6 abusive traffic are limited, it is envisioned that IPv6 will have an expanding role as an exploitation vector. In many cases, so long as an IPv6 traffic route exists, IPv6 traffic independence from IPv4 allows for different traffic paths and network policies that may be misconfigured or less securely due to little experience or use, such as those in Figure 2.1.

Specifically, a firewall might block outbound Transmission Control Protocol (TCP) port 25 traffic on IPv4, but not on IPv6. A great deal of research effort is needed to discover and repair the numerous attack vectors that exist within IPv6, hopefully, prior to discovery via a wide-scale victim exploitation event.

2.2 BGP Routing

A significant portion of our experiment relies upon examining BGP routing to determine if IPv6 spammers are exploiting BGP and whether they can be detected through BGP observations. Therefore, a brief look into BGP behavior is needed. In order for data to traverse from one end point to another, BGP is the preferred inter-domain protocol to route traffic between disassociated networks. BGP relies upon network boundaries, known as ASs, to exchange global traffic. Every logical part of the Internet exists within an AS, but each AS has various policies and preferred connections to other ASs that are determined by networking, economics, and politics. In order for ASs to communicate and advertise their

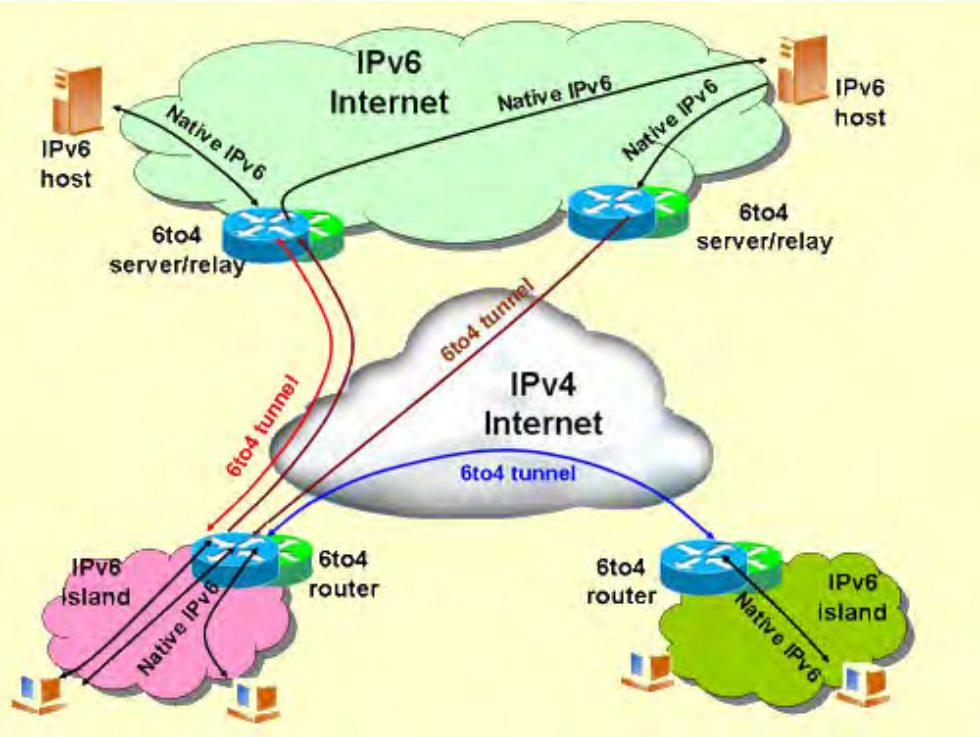

Figure 2.1: Example Network Displaying IPv6 Network Complexity [17]

services, BGP exchanges reachability information via Open and Update messages. Of specific importance is the Update message, as it allows BGP peers to send enough routing information to establish a graph of relationships between ASs in order to select the correct path for routing traffic [18].

To further study the behaviors of BGP, the Advanced Network Technology Center at the University of Oregon has created the RouteViews project. With RouteViews, any user is able to view historical routing information from BGP Update messages. RouteViews servers peer directly with other BGP routers to record BGP Update messages and store them into MRT formatted routing information base (RIB) files [19]. The RouteViews data repository effectively allows a user to return to a desired date and time and "replay" BGP routing behavior. The BGP replay ability afforded by RouteViews has been critical to the study of malicious traffic analysis by allowing the correlation of malicious traffic to unique BGP behaviors [20].

2.3 Related IPv6 Malicious Traffic Studies

In order to discover malicious traffic solutions in IPv6, it is vital to study and acknowledge previous areas of research that contain elements of possible answers to our research questions. For this study, we researched two key areas: spam and IPv6.

2.3.1 Spam

While there have been numerous spam-related studies, one study in particular is most relevant to our present research. Ramachandran and Feamster [20] studied several behaviors of network-level spammers and were able to successfully identify behaviors that aided in combating spam. Rachachandran *et al.* found that IPv4 sources of spam have a different distribution as compared to the sources of legitimate mail, that a small number of ASs account for nearly 40% of all of their measured spam, that most spam originated from Windows hosts, and that spammers used a technique known as BGP spectrum agility to remain untraceable [20]. Rachachandran's *et al.* lessons learned as a result of their studies consist of fully identifying the spam host for better spam filtering, using aggregate data to identify nefarious behavior, securing the Internet routing infrastructure, and combining network-level properties into trusted spam filters to aid in efforts to mitigate spam. Unfortunately, all of Rachachandran's *et al.* work was only conducted over IPv4 spam, which limits its application to IPv6 spam mitigation techniques. In contrast, this study seeks to perform several similar analyses to observed IPv6 spam traffic.

2.3.2 IPv6 Prevalence and Challenges

Until recently, there has been only a small amount of IPv6 traffic on the Internet. Dhamdhere *et al.* used BGP data to analyze the growth and performance of IPv6 on the Internet [13]. Dhamdhere *et al.* found that IPv6 is slowly growing each year, is more prevalent in Europe and the Asia-Pacific region, and that IPv6 performance measurements are comparable to IPv4 performance measurements as long as the AS-level paths are the same. When the AS-level paths differ, however, IPv6 performance can suffer in a drastic way. World IPv6 launch was regarded with optimism in the hopes that IPv6 awareness would bring in a large, new support for IPv6. While World IPv6 launch did have an impact, it was not as influential as desired [14] and more IPv6 topology and routing data are needed for comparison to IPv4 data.

2.3.3 IPv6 Client Adoption

Even though IPv4 addresses have been exhausted, a rapid adoption remains to be seen in wide-scale measurements of IPv6 clients. Zander *et al.* sought to quantify exactly how many clients connected to the Internet are using IPv6 [21]. Using Google ads to deliver a custom IPv6 capability test to web clients over a period of 10 months, Zander *et al.* established a dataset that demonstrated that while IPv6 capable Domain Name System (DNS) servers have increased 60% since IPv6-day 2011, client IPv6 adoption remains at a very low rate [21]. Even though the IPv6 client adoption rate remains generally low, more and more clients are still adopting IPv6, which would also allow native client resources to utilize IPv6. This is an important point, because if a client is IPv6 capable and infected with malware, that malware will most likely have an IPv6 communications capability. With the less-than-wide-scale adoption and the limited security strategies implemented in IPv6, a spammer sending malware or controlling a spamming botnet in IPv6 would most likely have more success since anti-spam efforts in IPv6 are still under significant development [11].

2.3.4 IPv6 and Spam

As they investigated why IPv6 was not being rapidly adopted as the preferred IP for MXs, Kosik *et al.* hypothesized that the persistence and sophistication of spam reduces the incentive of deploying IPv6 within a network [11]. Since the effectiveness of a DNSBL relies upon direct addressing and the reputation of a mail sender, Kosik *et al.* believed that the massive IPv6 address space would erode the DNSBL advantage. According to Kosik *et al.*, IPv6 DNSBLs would need to use a whitelisting method of filtration. That is, only IPv6 addresses known to be reputable are allowed through the mail filter. Due to the vast supply of IPv6 addresses, the benefits of caching DNSBL lookups become negligible, leading to an added latency on the side of the MTA, computation on the DNSBL host, and bandwidth to both. Otherwise a spammer can use a different IPv6 address with each spam and it is not computationally feasible to check every single possible IPv6 address against a DNSBL.

While IPv6 whitelisting may seem effective, it would discourage IPv6 connectivity across ASs and exhibit a tedious burden for the system administrator. As a result of IPv6 DNSBL problem complexity and the exploitation vectors afforded to spammers in IPv6, the more expensive, less effectual content-based spam filtering solution would be needed on IPv6

mail servers. The cost alone in employing content based MTA spam filtration could dissuade any organization from making the migration from IPv4 to IPv6. Kosik *et al.* had some well thought out ideas in regards to the possible effects of spam on IPv6, but few measurements were made within their study to substantiate their claims.

Another piece of research that sought to identify anti-spam approaches in IPv6 was the work by Rafiee *et al.* [22]. This research contains a great deal of background information on spam and the key characteristics of IPv6, while presenting some stimulating ideas on the security implications of spam in IPv6. For example, Rafiee *et al.* suggest that a spammer could use Bayesian poisoning techniques coupled with DNS DoS to bypass spam filtering in IPv6 networks. Rafiee *et al.* also identify the prodigious IPv6 address space as a spammer's resource and suggest that the IPv6 network prefix can be whitelisted in an internal network and that the Internet could use some sort of router prefix blacklisting method. The spam prevention methods discussed in this research have limited contributions, as some countries' Internet Service Provider (ISP)'s policies require periodic changes to routing prefixes, which has the second order effect of needing to also change router prefix blacklists.

CHAPTER 3:
Methodology

A common technique used to measure spamming behavior usually involves the use of a sacrificial target domain that entices spammers to send spam in order to conduct the desired experimental measurements. This type of target domain is often referred to as a "honeypot" or "honey domain," and it creates an appearance of a legitimate, exploitable target. While this is an effective experimental method when observing spamming behavior, there is little or no legitimate mail activity. The lack of legitimate email accounts often causes difficulty in inducing spammers to target the honeypot. To maximize the effectiveness of our experiment and further the study of real-world IPv6 deployment, we conducted our measurements using a production domain that is in active use. By using a live, corporate production domain with several thousand users, we were able to study non-random unsolicited mail activity over an extended period of time. This chapter will briefly discuss the experimental architecture used on our corporate production domain, which we will hereafter refer to as "`example.com`," but will also focus on our laboratory spamming testbed and our BGP correlation algorithm.

In addition to observing and collecting spam from a real-world production domain, we created a laboratory testbed environment that mimics the production domain's configuration. This testbed allows us to produce a comparative dataset and will hereafter refer to it as "`test.com`." The intent of our laboratory testbed is to re-create observed spamming behavior using various MTAs in order to fingerprint default MTA behavior that can be applied to metrics and used for analysis. While this MTA categorization is by no means all encompassing, it does provide some insight into default MTA behaviors and provides enough information to use as a MTA classification tool that can be applied to our `example.com` measurements.

The final piece of our experimental methodology relies on our BGP correlation algorithm, which replays our collected spam dataset from `example.com` against BGP Update messages in order to determine the state of the BGP routing table at the time the spam message was sent. By using the archive of BGP messages, we can determine the age of the BGP pre-

fixes corresponding to IPv6 spam origins to determine if a measurable relationship exists. Similar to Rachachandran *et al.*, we sought to identify whether spammers use BGP IPv6 spectrum agility, which is an obfuscation technique that consists of briefly announcing IP address space, usually hijacked IP addresses, from which to send spam and the routes to that IP address space once the spam has been sent [20].

3.1 Experimental Production Domain

The spamming measurements conducted on our live production domain constitute robust, initial efforts of our experiment. While a great deal of configuration, experimentation, and analysis has been conducted on our `example.com` measurements, that work was largely performed by our research colleagues. Instead, the focus of this thesis relies on the setup and validation of our laboratory spamming testbed and tying in our `example.com` results with known BGP spamming trends. However, we review the configuration of the experimental production domain here to provide the relevant background on our subsequent analysis.

3.1.1 Configuring `example.com`

The first step in designing the experiment was selecting a protocol that would enable measurement of abusive IPv6 traffic within a production domain. For our purposes, we selected SMTP because it has built-in failover capability to facilitate using IPv6 when IPv4 connections are deemed abusive, is highly configurable, and is a known choice for spamming activity [23]. Next, we created a dual-stack, tertiary MX server to act as our spam sensor. With the sensor in place within our production domain, we created configuration parameters that would still allow legitimate mail flows through our primary (smtp1) and secondary (smtp2) MXs while directing known spammers or less favorable mail flows to our spam sensor, as demonstrated in Figure 3.1.

Further configuration metrics regarding `example.com` will be explained in Section 3.2, which had a purpose of mimicking the spamming behaviors seen within our large scale production domain.

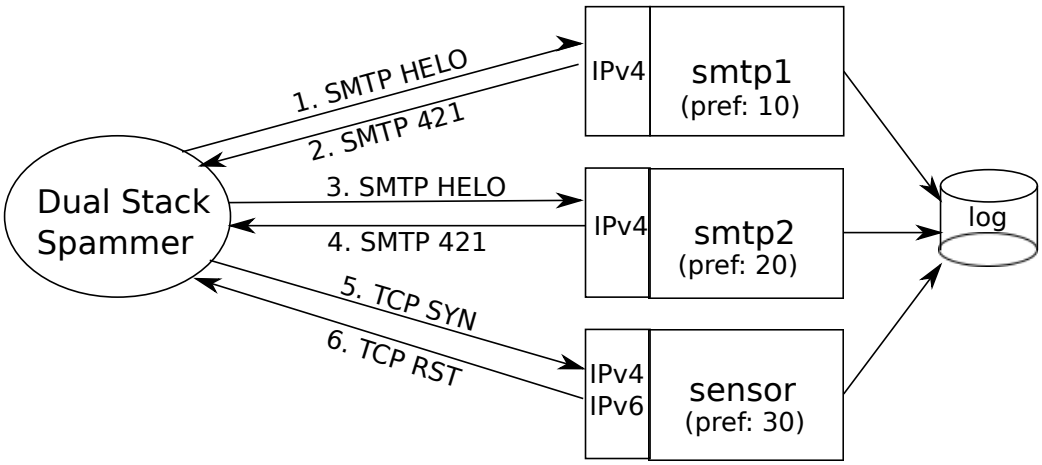

Figure 3.1: `example.com` Experimental Setup

3.1.2 `example.com` Dataset

To properly record all measured spam events from `example.com`, we collected packet capture file (pcap) traces of all incoming TCP connection attempts, both IPv4 and IPv6, over port 25 at the sensor MTA for nearly 11 months, January through November 2013. Each pcap included timestamps, IP, and TCP information of spamming connections, in addition to using the following methods to enumerate each spam attempt:

- *Autonomous System Number (ASN)* for the source IP, using Team Cymru's IP-to-ASN lookup [24] (for 6to4 IPv6 addresses [25] we used the ASN associated with the IPv4 address of the 6to4 gateway, which is embedded in the IPv4 address).
- *Reverse Domain Name System (RDNS)* name for the source IP, obtained by performing a `PTR` lookup on the IP.[1]
- *OS* version and flavor of the sending MTA, as interpreted by passive operating system fingerprint (p0f) [26].
- *Reputation* data for the sending IPv4 address from two sources: CYREN [2] and SpamCop.[3]

These spamming record data fields acted as a driving force in the development of our experimental spamming testbed. Through observation and analysis of p0f records, we deter-

[1] DNS Queries: Reverse DNS Lookup and PTR Query [http://www.dnsqueries.com/en/reverse_lookup.php/]
[2] CYREN Security Services: Embedded IP Reputation [http://www.cyren.com/]
[3] SpamCom Reporting Service [http://www.spamcop.net/]

mined the most frequently used OSs by spammers in order to use same OS in our test.com experiment. Finally, we used each source IP address and its measured example.com spamming timestamp to run during each instance of our BGP correlation algorithm, which will be further explored in Section 3.3.

3.2 Laboratory Spamming Testbed

To properly establish the context of test.com, we must review in detail its configuration, which mirrors the configuration used in example.com. This section will explain how test.com was configured, exchanged email, and conducted measurements to those similarly seen in example.com.

3.2.1 Experimental Design

A domain is configured to exchange email through the use of MX records in the DNS, with each MX record having an assigned target mail server with a designated name (e.g., TMS1.test.com) and numeric preference value (e.g., 10). Using SMTP, the delivering MTA attempts to send mail to the servers with the lowest (most preferred) preference value. The overall purpose of the entire experiment was twofold: to determine MTA spamming behavior given both IPv4 and IPv6 as available transport protocols and to understand how large a role the spamming MTA's OS plays in spamming behavior. We therefore needed to ensure that network protocol selection and preference was followed. We therefore relied on SMTP to operate as designed [23], [27]–[29] and on the MX preference of the MTA names and their corresponding IPs [23], [27]. To link a mail server's hostname to an IP address, DNS name resolution resolves the MX record to the A record (A) or quad-A record (AAAA) record delivered by the IPv4 or IPv6 protocol. The specific protocol used is determined by the existence of an A or AAAA in the DNS, the network availability of the sending server, the sending server's local policy, and the OS used by the sending MTA.

Our testbed DNS contained two MX records that had targets corresponding to the primary (TMS1.test.com) and secondary (TMS2.test.com) MTAs, both of which were configured as IPv4 MTAs only (i.e., the primary and secondary MTAs had no IPv6 addresses). The primary and secondary MTA each had an A record and rejected incoming mail requests from all IP addresses. The rejected mail message consisted of a SMTP 421 error code ("service not available") that would follow after the SMTP HELO command from the sending MTA.

This rejection was achieved through a script that ran on TMS1 and TMS2 that would attain that same result if a sending IP to example.com correlated to an IP address contained within the DNSBLs or reputation policy.

Note that this default rejection behavior is different than used in the production example.com setup; instead we wish to redirect all traffic in the testbed for the purposes of understanding individual MTA behavior.

Name	Type	Pref.	Target
test.com.	MX	10	TMS1.test.com.
test.com.	MX	20	TMS2.test.com.
test.com.	MX	30	TMS3.test.com.
TMS1.test.com.	A		192.0.2.1
TMS2.test.com.	A		192.0.2.2
TMS3.test.com.	A		192.0.2.3
TMS3.test.com.	AAAA		2001:db8::3

Table 3.1: DNS Resource Records Corresponding to test.com.

We added a third MX, having IPv4 and IPv6 connectivity, a higher MX preference value, and pointing to TMS3.test.com, a dual-stack server with both A and AAAA records. The testbed's DNS configuration is shown in Table 3.1. We refer to our third MX server as TMS3. TMS3 always rejects incoming connection attempts to port 25 over IPv4 and IPv6 by issuing TCP RST packets in response to the TCP SYNs.

Thus, to simulate the rejection behavior of DNSBLs and to minimize the misuse of network resources, we issued SMTP 421 error codes, which exist in the application layer, in response to connections at TMS1 and TMS2. For rejections at TMS3, we issue the TCP RST at the transport layer, ensuring that any connect attempt is actively rejected and understood by the sending MTA. For example, given our experimental setup demonstrated in Table 3.1 and Figure 3.2, a spammer attempting to send mail to anyone at test.com would first try TMS1, then TMS2, then TMS3.

The purpose of the TCP RST for each spam attempt at TMS3 was to understand default network behavior of commonly used OSs and MTAs as a baseline for comparison with observed behavior on the production domain.

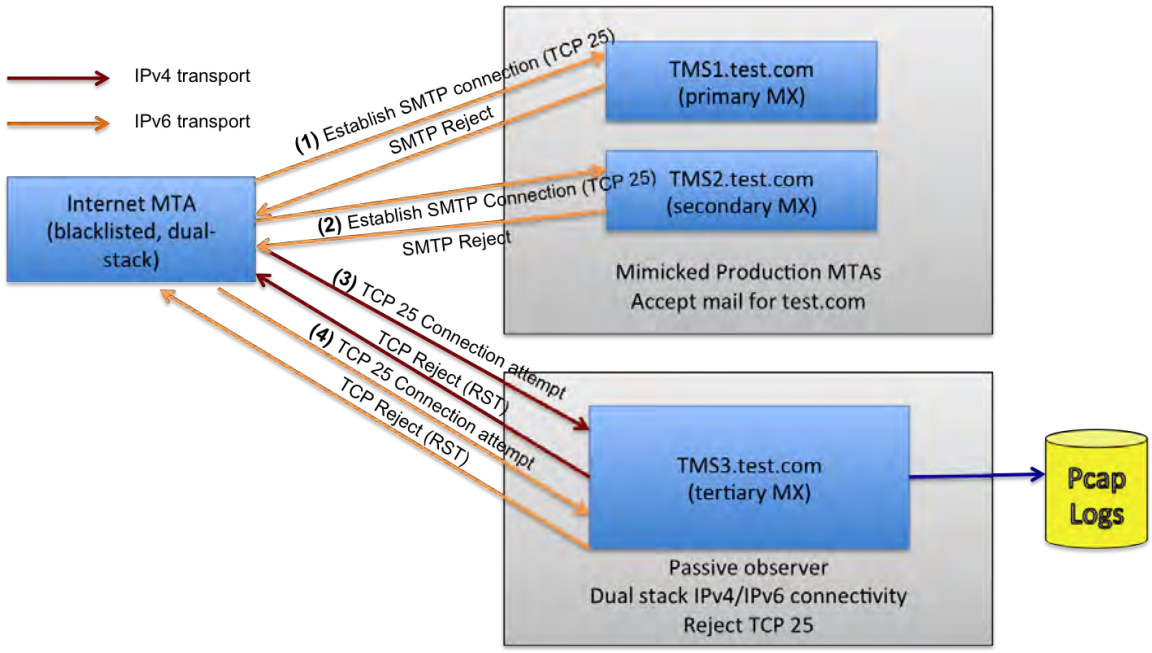

Figure 3.2: `test.com`'s Operational Behavior

3.2.2 Testing MTAs

We configured three different dual-stack "spammers" utilizing well known MTAs: Microsoft Exchange [30], Sendmail [31], and Postfix [32], as shown in Table 3.2. In each iteration of our spam testbed experiment, we recorded each MTA message delivery attempt to an email address within our test domain for a measurement period of 36 hours. Of note, each machine within `test.com` used a global IPv6 address, suggesting that the sending OS would prefer the IPv6 address over the IPv4 address [29].

Name	Operating System	MTA Software
`spam1.private.com`	`MS SBS 2011`	`ME 2010 14.01`
`spam2.private.com`	`Ubuntu 12.04`	`Postfix 2.9.6`
`spam3.private.com`	`Ubuntu 12.04`	`Sendmail 8.14.4`

Table 3.2: Test Domain Configuration

For measurement purposes, we differentiated application-level TCP-level connection attempts by each spammer as seen at `TMS3`. We defined a maximum time between `SYN` packets as 20 seconds, and grouped `SYN` packets with the same TCP source port and source IP

address. We refer to the resulting group as a *connect attempt*, referring to the socket call by the same name to produce that behavior. Our goals in conducting this experiment were three-fold. First, we wanted to characterize and explore each MTA's attempts to establish connections over IPv4 and IPv6 at TMS3, over time. Second, we wanted a set of control data to which we could compare our measurements observed on the production domain. Third, we wanted to understand how each sending MTA chose port numbers while making repeated attempts to send mail over time.

Some of the specific questions we sought to answer with this testbed are the following:

- For each connection retry by the MTA, does the connection try both IPv4 and IPv6, and which does it try first?
- How much time elapses between IPv4 and IPv6 connection attempts associated with each spammer "retry?"
- How many SYNs are sent for each observed connect attempt, and how much time elapses between each connect attempt?
- How many distinct connect attempts are observed over an extended period of time?
- What is the port usage behavior observed with associated IPv4 and IPv6 connect attempts?

Of particular interest in our experiment was the behavior of port selection and how OS and MTA choice affected the port numbers used. According to IANA, port numbers that range from 49152-65535 are titled "dynamic" or "ephemeral" ports. Port numbers in this range are set aside for dynamic use and are predesignated as being unable to be assigned by IANA as system or user ports. Proper ephemeral port usage policy calls for an application to select any port available within the range, so long as the application does not assume or identify a specific port number to be used consistently [33]. Ephemeral port number selection play an important role in our experiment if we are to successfully identify sending MTA characteristics. Determining whether an OS or MTA uses random, sequential, or cycled ephemeral ports for each connect attempt, as well as what base port offset is used, was a significant part of mapping observed behaviors to OSs. Using these feature, we seek to establish a measurable baseline of commonly used mail applications and OSs. The results of our test.com experimentation, found in Table 4.3, will be further discussed and analyzed in Chapter 4.

3.3 BGP Correlation Algorithm

As noted by Rachachandran *et al.*, some sophisticated spammers are exploiting the weaknesses of the Internet routing infrastructure via short-lived BGP route updates from hijacked prefixes [20]. While their work identified BGP spectrum agility as a unique and measurable spamming behavior, their work was focused only on IPv4. We sought to expand their efforts to IPv6 to see if, in fact, IPv6 spammers were using the same BGP spectrum agility behavior to complement their spamming efforts. As previously mentioned, we relied on RouteViews to provide all BGP updates for comparison to our `example.com` spamming measurements. To fully comprehend BGP spectrum agility, we must first understand what it means to "hijack" a BGP prefix.

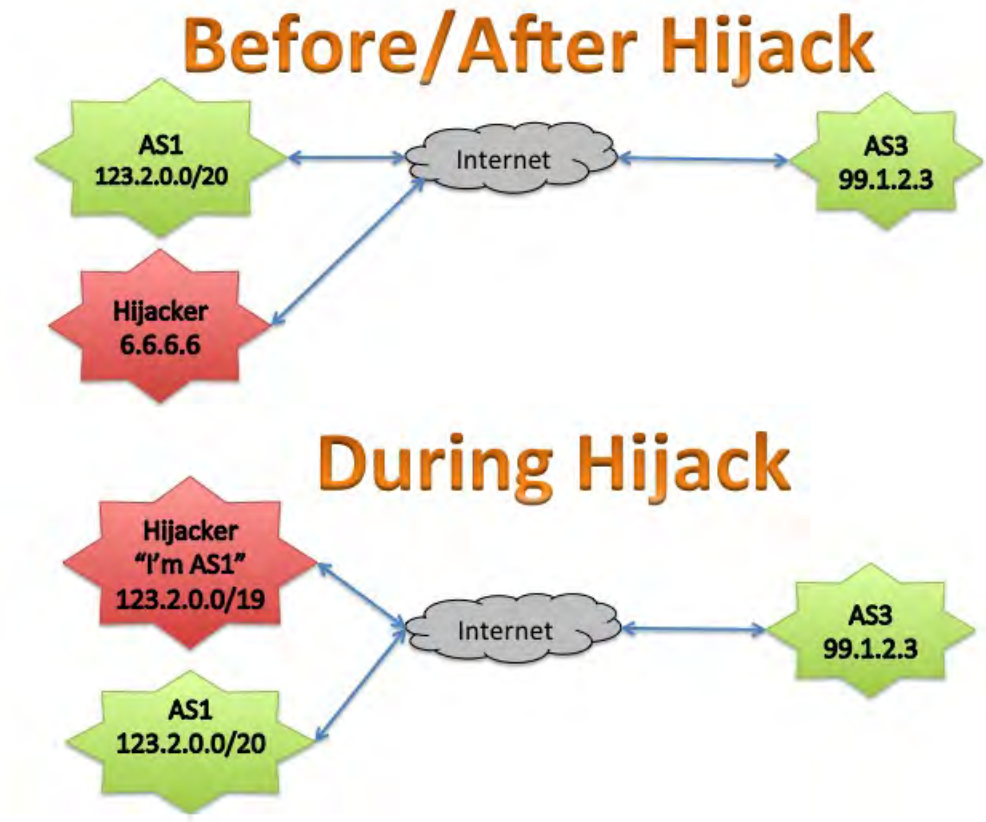

Figure 3.3: Example BGP Route Hijacking

When BGP is used to route traffic between AS, it relies upon advertised AS routes to send traffic from source to destination AS. The design of BGP allows each AS to implicitly

trust a peered AS, which allows a malicious person to masquerade as a peered AS and advertise a new route. This spoofing of a newly advertised BGP route is known as BGP hijacking 3.3. The hijacker then illegally selects a block of IP address space that does not belong to them and advertises the IP address block as a spoofed AS route. Once the attacker advertises this spoofed route, the attacker appears to come from an IP address that does not belong to them and their traffic appears to be routed from the spoofed, remote AS. When the attack is complete the attacker sends an AS withdrawal message to restore the AS path advertised prior to attacker interference. Malicious persons using this technique often employ unused network addresses, known as "dark net" addresses, to reduce the likelihood of the IP address ending up on a DNSBL. Figure 3.3 depicts an example of a prefix hijacking and is a proven method that spammers use to obfuscate themselves in order to avoid spam filtration mechanisms [20].

3.3.1 `bgpdump`

RouteViews stores their BGP update messages in a RIB file in the popular MRT format, which is a binary formatted file. In order to read each RIB, we relied on a well known BGP tool called `bgpdump`.[4] `bgpdump` is able to convert each RIB file to human-readable, ASCII text then parse each line of the file in order to allow the user to query the desired BGP data. For the purposes of our algorithm, `bgpdump` successfully parsed the timestamp, message type, source IPv6 address network prefix, and the origin AS of each BGP Announce or Withdrawal message.

3.3.2 Algorithmic Description

Once we have the parsed data from `bgpdump`, we use each instance of recorded spam from our `example.com` experiment and search all BGP updates from the time of one instance of spam to the next instance of spam time. Our algorithm identifies any IPv6 network prefix BGP announcements or withdrawal messages that have an exact match to the recorded IPv6 spamming IP address and records some other statistical data.

[4] BGPDump Repository [https://bitbucket.org/ripencc/bgpdump/wiki/Home/]

```
tree = new(Radix trie)
spamtime.previous = 0
for each spam in database do
    spamtime.next = timestamp.spam
    replayBGPupdates(spamtime.previous, spamtime.next)
    addspam(spam, tree)
end for
```

Figure 3.4: BGP Correlation Algorithm Pseudocode

The pseudocode seen in Figure 3.4 may be straightforward, but the elements needed for algorithm execution were sufficiently complex and will be further explored, along with the results, in Chapter 4.

CHAPTER 4:
Experimental Results

In the previous chapters we introduced our experimental design, which involved measurements from a real-world production domain, a simulated spamming domain, and a BGP correlation algorithm. In order to bring this large corpus of information together this chapter will focus on the results and analysis from each experiment. Our dataset suggests that while there is a slowly increasing amount of IPv6 activity on the Internet, it is still orders of magnitude less than its IPv4 counterpart. Additionally, spammers will continue to use proven IPv4 exploitation methods, such as BGP spectrum agility, when sending spam in IPv6.

4.1 Notable `example.com` Analysis

One of the aims of the `example.com` experiment was to associate an IPv4 address with an IPv6 address from dual-stacked spammers. A mechanism to accurately identify a spammer based solely on sent spam traffic has yet to be developed, but a few forensic details from the spam traffic can allow us to correlate an IPv6 address to the likely IPv4 address for spam attribution. The time between each IPv4 and IPv6 `connect` attempt, embedded IPv4 host addresses in IPv6 addresses, ASN use, DNS `PTR` records, TCP source port proximity, IP address proximity, and overall `connect` behavior were all analyzed in the `example.com` experiment. Once each of those connection statistics were recorded, they were scored based on a naïve matching algorithm that utilized a highest confidence score to associate an IPv4 and IPv6 address. The matching algorithm was developed outside of the thesis as part of a collaborative work effort and is subject of a separate publication under review. However, utilizing the `test.com` experimental results, we verify the matching algorithm through the understanding of actual ground-truth behavior in our laboratory testbed.

While there were a myriad of statistics developed as a result of `example.com` measurements, as seen in Table 4.1, there were some key points. First, 80% of the IPv6 addresses observed were associated with IPv4 addresses, which made up 87% in total of all observed IPv6 addressees within the experiment. Second, all but 1% of associated IP addresses used the same ASN, which can help in discovering spammers that use BGP route prefix

hijacking. Third, the Windows hosts sending spam tended to select source ephemeral port numbers within a small range of only 500 different ports from the initial source port number for each `connect` attempt. This behavior was in direct contrast to Linux and other OS MTAs, and when combined with the results of the tool p0f, created a unique profile of a spamming Windows host.

Statistic	**Count** (%)
Unique IPv6 addresses	3670
- IPv6 assoc.	2907 (79)
- IPv6 assoc. – first rogue	575 (15.7)
- IPv6 assoc. – single IPv4	2722 (74.2)
- IPv6 unassoc. – 6to4 w/ embedded IPv4	42 (1.1)
IPv6 `connect` attempts	135770
- With IPv4 association	118177 (87)
- Close proximity assoc.	85640 (63)
- Rogue w/ previous close proximity	32537 (24)
IPv6 Associations	3319
- Embedded IPv4 host	269 (8)
- Embedded IPv4 host (6to4)	215 (6.5)
- ASN	3281 (99)
- PTR	309 (9.3)
- OS	1823 (55)
- OS mismatch	40 (1.2)

Table 4.1: Summary of Results from IPv6/IPv4 Address Association.

Once IPv4 and IPv6 address association was accounted for, a holistic analysis was conducted on IPv6 spamming attempts over the course of the entire experiment. As seen in Figure 4.1, there were roughly two orders of magnitude more attempts in IPv4 than IPv6 with an average of 2908 IPv6 `connect` attempts per week. Of the IPv6 spammers, 20% attempted to connect to the sensor MTA only once, 70% up to 10 times, and 5% more than 100 times. Another interesting trend was also observed, see Table 4.2, in the OSs of spamming hosts using IPv4 and IPv6. The various results from this experiment led to the following conclusions: that IPv4 spamming activity is still about 100 times more pervasive than IPv6 spamming activity, that the experiment was unsuccessful in determining the exact profile of any IPv6 spammer measured in the experiment, and the amount of IPv6 traffic did not increase over the course of the experiment.

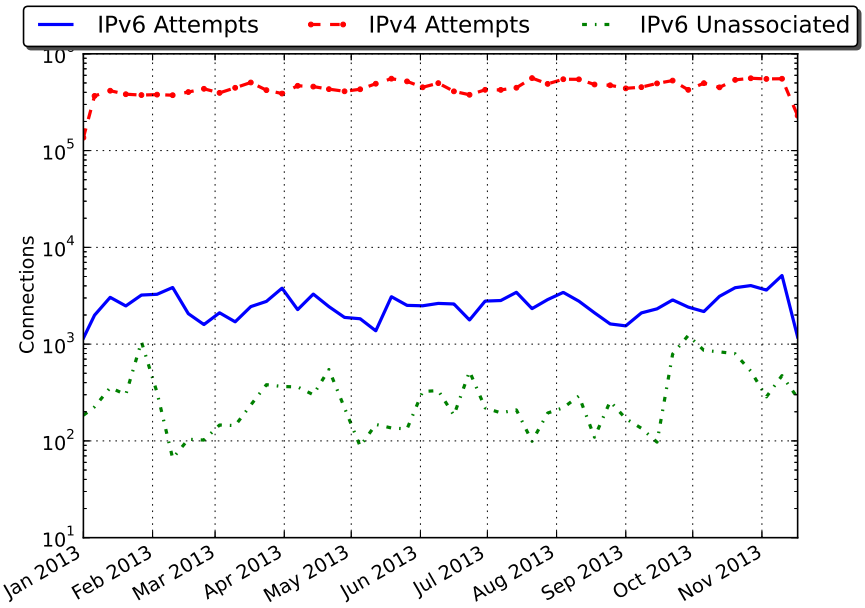

Figure 4.1: Weekly `connect` Attempts to `example.com`'s Sensor MTA.

OS	IPv4 Hosts	IPv4 Attempts	IPv6 Hosts		IPv6 Attempts	
			Assoc.	Unassoc.	Assoc.	Unassoc.
Windows	593150 (62.41)	9109137 (49.77)	293 (8.40)	105 (13.76)	18492 (13.43)	2652 (15.07)
- Windows NT	46027 (4.84)	1928541 (10.54)	13 (0.37)	4 (0.5)	77 (0.06)	28 (0.16)
- Windows XP	285454 (30.04)	4355433 (23.80)	2 (0.06)	2 (0.3)	12 (0.01)	12 (0.1)
- Windows 7	261669 (27.53)	2825163 (15.44)	278 (7.97)	99 (12.8)	18403 (13.37)	2612 (14.8)
Linux	66876 (7.04)	7025690 (38.38)	1900 (54.46)	317 (41.55)	96976 (70.43)	11842 (67.31)
- Linux 2.0	7 (0.00)	21788 (0.12)	-	-	-	-
- Linux 2.2	7353 (0.77)	339103 (1.85)	52 (1.49)	26 (3.4)	1112 (0.81)	271 (1.5)
- Linux 2.4	7367 (0.78)	723900 (3.95)	163 (4.67)	21 (2.8)	3580 (2.60)	273 (1.6)
- Linux 2.6	29431 (3.10)	4464112 (24.39)	492 (14.10)	115 (15.1)	30605 (22.23)	6364 (36.2)
- Linux 3.x	22718 (2.39)	1476787 (8.07)	1193 (34.19)	155 (20.3)	61679 (44.79)	4934 (28.0)
Solaris	198 (0.02)	24781 (0.14)	-	-	-	-
Mac OS X	999 (0.11)	128024 (0.70)	-	-	-	-
FreeBSD	2066 (0.22)	261686 (1.43)	-	-	-	-
OpenBSD	20 (0.00)	416 (0.00)	-	-	-	-
Other	145 (0.02)	19232 (0.11)	7 (0.20)	3 (0.4)	64 (0.05)	9 (0.1)
Unknown	286902 (30.19)	1734496 (9.48)	1289 (36.94)	338 (44.3)	22163 (16.10)	3090 (17.6)

Table 4.2: `example.com`'s Inferred Operating Systems for Spamming Hosts

4.2 MTAs Behaving Badly

While the experimentation and analysis conducted with `example.com` was extensive, we felt that it would be advantageous to understand the default network behaviors of com-

monly used OSs and MTAs as a baseline for comparison with the observed behavior in example.com. As previously mentioned in Section 3.2, we sought to explore how a MTA preferred each IP version, port numbers, and to identify default behaviors. To determine a common behavior for each MTA and mimic the sending of spam, we observed how each MTA would attempt to send a single message over a period of 48 hours to a sensor MTA. Our sensor MTA would continually send a TCP RST for each attempt by the sending MTA to deliver the "spam" message, which mirrored the behavior of example.com 's sensor MTA 3.1. Once we parsed and analyzed the pcaps from our experiment, we generated Table 4.3. These results were instrumental in identifying anomalous behavior from example.com.

Software	**Sendmail**	**Postfix**	**MXS 2010** (IPv6)	**MXS 2010** (6to4)
IPv6 or IPv4 first?	IPv6 (100%)	IPv4 (62%) IPv6 (38%)	IPv4 (100%)	IPv4 (100%)
Time per connect attempt?	10 min	83 min, 70 min thereafter	10 min	10 min
SYNs per connect attempt?	1	1	3	3 to 6
Port selection?	Ephemeral	Ephemeral	Ephemeral	Ephemeral
Port range (IPv4)?	49224-50091	42502-57419	6331-65741	7166-65079
Port range (IPv6)?	57626-58493	35469-35603	6334-65473	7170-63309

Table 4.3: Results from test.com Spamming Experiment

While we sought to correlate results from test.com to the observed behaviors of example.com, it is important to note that behaviors can be the result of either the application or the OS. For example, the number of SYNs per connect attempt is believed to be an OS specific behavior. Furthermore, there could be conditions within each OS that make behavioral association of the example.com dataset complicated. We were unable to make an attributable claim when comparing example.com spam behavior because we have no way of knowing whether the spamming behavior was the result of native OS procedures, an OS specific MTA, or an OS infected with spamming malware.

4.2.1 Sendmail v8.14.4

When comparing all three MTAs, Sendmail exhibited the most expected network preference characteristic of an MTA in that it always preferred IPv6 over IPv4. With each connect attempt observed, we detected that IPv6 was always tried prior to IPv4. We observed only one connect attempt per selected IP. The time difference observed between the IPv6 connect attempt and the IPv4 connect attempt that followed was less than half a second and should be considered inconsequential. Each SMTP connect attempt consisted

of only one IPv4 and one IPv6 connection and the first retry attempt occurred 3 minutes 49 seconds after the first SMTP `connect` attempt. Following the first SMTP connection retry, Sendmail attempted to connect every 9:55 seconds. Sendmail chose an ephemeral port for the first SMTP `connect` attempt, one for IPv4 and one for IPv6. Each subsequent `connect` attempt saw an increase in the port number by some value between three and seven, but throughout the entire transaction, it can be observed that the IPv4 and IPv6 port numbers stayed in range of their initial ephemeral port number. While this port selection behavior will change with each new email, we observed that Sendmail incrementally increased the port number with each email retry, beginning with an arbitrary source port.

A particular observation from the `example.com` dataset regarding the behavior of the Sendmail MTA, was the fact that Sendmail would periodically retry to send mail from the same source IP address and port minutes, hours, and days after initial `connect` attempts. This behavior was a particular incident observed in the `example.com` dataset that we did not reproduce in `test.com`, so we are unsure as to how widespread the behavior is within real world production domains. After further investigation and data correlation in the `example.com` dataset, we were able to determine that this anomalous behavior of same source IP and port reuse was conducted by a single organization. We were able to confirm this relationship by directly contacting the organization that originally sent the mail to determine which MTA was currently being employed within their organization. We then looking for this pattern of behavior in our `test.com` dataset and we unable to find any instance of same source IP and port re-use. Unfortunately, we were unable to account for this behavior in every instance of `connect` attempts over extended time periods observed in the `example.com` dataset and we are unsure of the Sendmail configuration that caused this unique behavior.

4.2.2 Postfix v2.9.6

By observing our Postfix measurements within our MTA testbed, we can say that Postfix comparatively performed in the most random way. Postfix tried both IPv4 and IPv6 with one `SYN` during each `connect` attempt and attempted IPv4 half a second before attempting IPv6. The time period that elapsed between each SMTP `connect` attempt was the most irregular of the three MTAs. The next SMTP `connect` attempt following the first was at 7:50 seconds, followed by a 10 minute, 20, and 40 minute retry attempt. Following the 40 minute retry interval, Postfix reattempted an SMTP `connect` every 10 minutes there-

after. Postfix exhibited the same port behavior as Sendmail by choosing an ephemeral port to connect to for IPv4 and IPv6, but the two MTAs did not choose the same ephemeral ports. Following the first SMTP `connect` attempt's port number, each reattempt saw the port number increased by some value between three and seven due to other source port allocations created in the interim.

The Postfix results regarding IP preference selection was particularly interesting. According to Postfix documentation, IPv6 is preferred over IPv4 when both protocols are available and, in versions 2.8 and later, allow the administrator to specify which protocol should be preferred [32]. When both protocols are available, IPv6 should be used first by default but a disclaimer in the documentation informs readers that default settings are "unsafe." If mail delivery is attempted when there is an IPv6 outage on Postfix, the message could fail to deliver even if a route still exists using IPv4[5]. Also, if the protocol preference is set to "any," there will be an equal preference for both IPv4 and IPv6, making the selection non-deterministic. Finally, we observed the same random protocol preference selection in both `example.com` and `test.com`, which assisted and validated our IP address association from `example.com`.

4.2.3 Microsoft Exchange 2010 v14.01

Microsoft Exchange Server 2010 (MXS) exhibited the most identifiable MTA behavior of all our measurements. Two different types of experiments were conducted with MXS, the first using native IPv6 connectivity and the second using the 6to4 tunneling protocol. MXS makes `connect` attempts on both IPv4 and IPv6 for the initial and subsequent connections, with IPv4 tried first. For each `connect` attempt, MXS sends three `SYN`s for both IPv4 and IPv6 and each `connect` attempt are within a half of a second of each other. After the initial SMTP `connect`, another was launched 14 minutes later. Following the 14 minute period, a `connect` was launched every 10 minutes thereafter. During each IPv4 and IPv6 `connect` attempt, an ephemeral port was chosen by the OS for the IPv4 `connect` attempt and then the port value increased by one for the IPv6 `connect` attempt. This ephemeral port and its plus one counterpart was observed at each SMTP `connect` attempt. Overall, the triple `SYN`s over each `connect` attempt with IPv4 and IPv6 and the adjacent port numbers for each `connect` attempt are indicative of MXS behavior and could be a signature of a spammer

[5]Postfix Configuration Parameters [http://www.postfix.org/postconf.5.html]

utilizing MXS.

When evaluating each MXS spam attempt form `test.com`, there was very little discernible difference between the native IPv6 and the 6to4 results. Both methods, native IPv6 and 6to4, tried IPv4 prior to IPv6 for each `connect` attempt and utilized the same time retry period. Also, the same ephemeral port selection behavior was present in that the port values increased during each `connect` attempt and each subsequent `connect` selected an random port number. The only observable difference that can be seen was the fact that 6to4 employed between three and six `SYN`s for each `connect` attempt while native IPv6 connectivity uses only three `SYN`s for each `connect` attempt. Finally, we must mention the lack of well-behaving protocol preference conformity observed by MXS [29]. Prior to each experiment, we ensured that both the Windows Server running MXS and MXS itself preferred IPv6 over IPv4 through the use of the 'netsh' command line tool[6]. Regardless of the prefix policy settings, every `connect` attempt utilized IPv4 first. This behavior suggests that MXS is a badly behaving MTA that ignores protocol preference established by the OS at the application layer.

Finally, we analyzed the trends from `test.com` dataset and applied what we learned to the `example.com` dataset. Unfortunately, correlation between IPv4 and IPv6 addresses from `example.com` is a difficult and ongoing discovery process, so we were unable to quantify the use of IPv4 or IPv6 first as a characteristic as in Table 4.3. We were able to successfully determine a few other unique behaviors that allow us further insight into profiling our observed `example.com` spam behavior. Whether or not a `connect` attempt used a single `SYN` or a sequence of three `SYN`s, the IP used for each attempt, and the port numbers were all analyzed.

Of note, 14% of the `example.com` spam data included three or more `SYN`s per `connect`, which parallels with our observed `test.com` spamming behavior of MXS. While this is certainly a highly correlative behavior, it is unknown whether that is because the spam senders are actually MXS MTAs, other MTA implementations on Windows machines, or Windows malware. Of note, all port numbers from `example.com` were ephemeral, which also parallels with our spamming testbed dataset. While this additional analysis of the

[6]To view prefix policy: 'netsh interface ipv6 show prefixpolicies'
To change prefix policy: 'netsh interface ipv6 add prefixpolicy address precedence-value label'

Total Packets	109138752
SYN Count	28898011 (IPv4) 1315838 (IPv6)
connect attempts	20693161 (<3 SYNs/attempt) 3478195 (>3 SYNs/attempt)
Port Range	1024-65519

Table 4.4: Statistics from example.com Dataset

example.com dataset displays strong related behaviors with the test.com measurements, it is by no means a definitive answer regarding default spamming behavior and is more of a demonstration of ground truth validated by a realistic production dataset.

4.3 Tracking IPv6 Spam via BGP

Address Type	**Type Count**
2001::	129
2002::	2
2400::	9
2600::	88
2a00::	237
Network Prefix	**Count**
/16	2
/21	1
/26	56
/29	6
/32	291
/36	3
/40	4
/44	2
/48	100

Table 4.5: BGP Algorithm Statistics from BGP Episodes Under 25 Hours

In Section 3.3, we introduced our BGP Correlation Algorithm that we developed in order to determine if spammers were using BGP spectrum agility in IPv6 [20]. Our intent was to develop an algorithm that replayed BGP update messages before, during, and after each recorded instance of spam from example.com. A detailed version of our algorithm can be studied in the Appendix. Using the IPv6 network prefix within a BGP route announcement or withdrawal and an example.com IPv6 spam address, we wanted to observe instances of

a BGP announcement containing a large IP address space, a recorded IPv6 spam attempt from `example.com` with a timestamp closely following the BGP route announcement, and then a BGP withdrawal of the route following the last recorded IPv6 spam attempt. We will hereafter refer to this sequence of events as a BGP episode. The discovery of the BGP spectrum agility technique was initially made in IPv4 spam and although there were only a few spamming entities that seemed to use this technique, there were some very unique, consistent, observable behaviors associated with BGP spectrum agility [20]. We observed some instances of behavior in IPv6 BGP messages that mimics that of BGP spectrum agility.

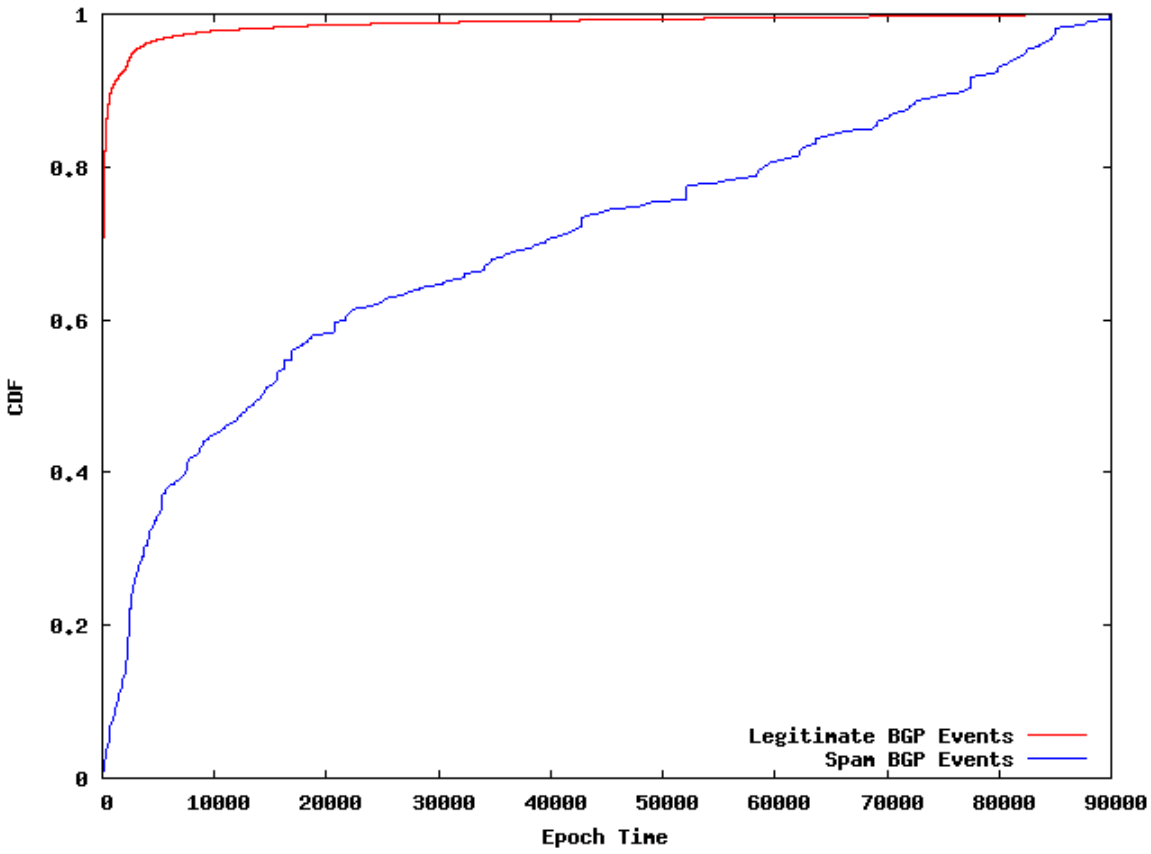

Figure 4.2: Length of Short-Lived BGP Episodes Lasting Under 25 Hours

We observed some 465 instances of BGP announce and withdrawal updates, with a lifetime of less than 25 hrs, that correlated with recorded IPv6 spam attempts, such as the BGP episode displayed in Figure 4.3. The spam attempts listed in Figure 4.3, originating from `2a01:111::/32` , displayed 11 instances of BGP agility behavior lasting less than 25

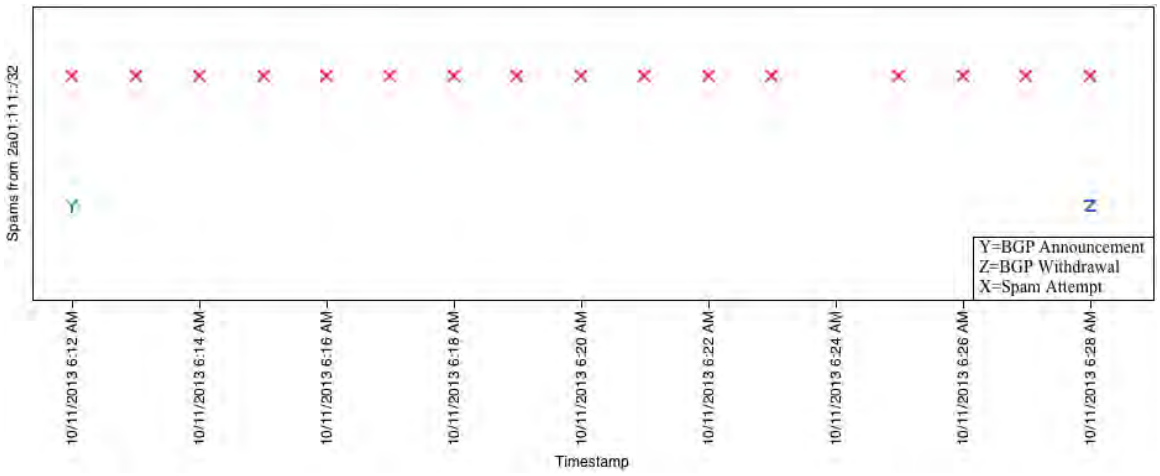

Figure 4.3: Observation of `2a01:111::/32` BGP Agility Behavior

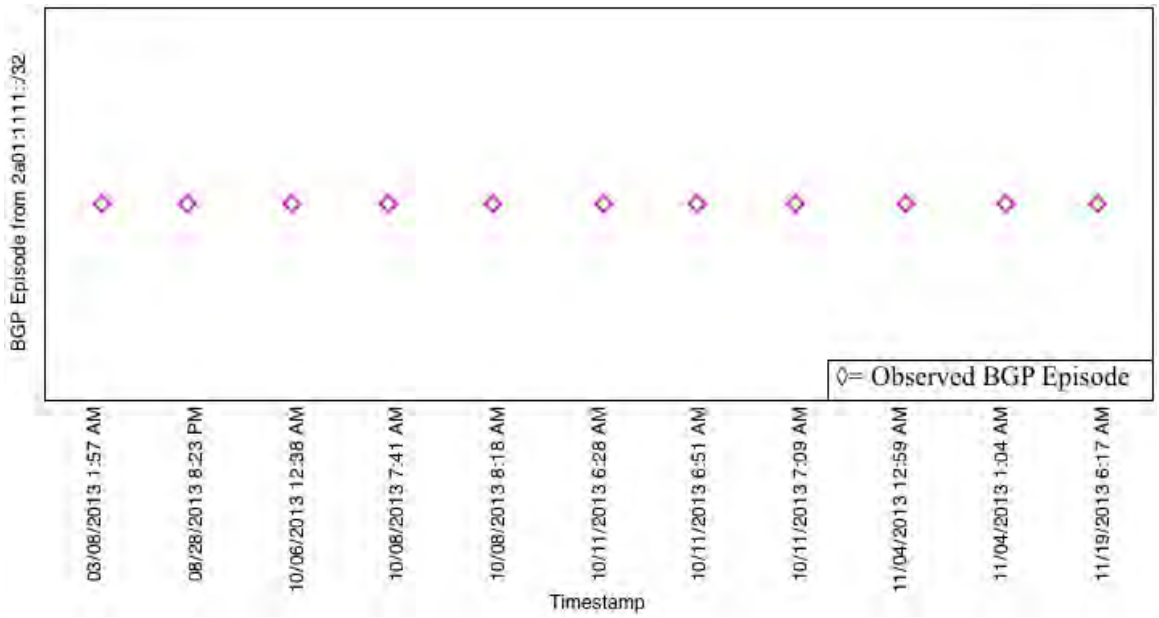

Figure 4.4: `2a01:111::/32` BGP Agility Episodes Lasting Less Than 25 Hours

hours. Most of the BGP episodes from `2a01:111::/32` occurred in October 2013 and on October 11, 2013, there were three BGP episodes, which is highest number of instances of BGP episodes recorded by `2a01:111::/32`. It is important to note the 25 hour lifetime of the observed BGP agility behavior, as seen in Figure 4.2, as this was the threshold that we set for our area of study. Anything larger than a 25 hour lifetime starts to introduce more likelihood that we inappropriately associate a BGP updates message as BGP spec-

trum agility behavior. We believe that a misconfigured border router or a poorly managed ISP could be falsely flagged as a spamming entity in our dataset without any limitation on the lifetime threshold to determine BGP spectrum agility. In evaluating Figure 4.2, we can see that nearly 60% of all the recorded spam associated with BGP agility behavior lasted less than 20830 seconds or 5 hours and 47 minutes. As seen in Table 4.5, the IPv6 address types include `2001`, `2002`, `2400`, `2600`, and `2a00` and the prefixes include `/16`, `/21`, `/26`, `/29`, `/32`, `/36`, `/40`, `/44`, and `/48`. These IPv6 prefixes are not representative of observations made by Ramachandran *et al.* because Ramachandran *et al.* observed IPv4 addresses, which came from IPv4 address space and is roughly 17 billion times smaller than IPv6 address space. For example, while ISPs in IPv4 are typically expected to have a /16 network, ISPs in IPv6 might be expected to have a /32 network. The differences in the network sizes observed in our BGP spectrum agility data are not uncommon due to the large, less-trafficked IPv6 address space [9] and the wide availability of unallocated IPv6 networks.

While a majority of the IPv6 prefixes list in Table 4.5 can be observed in the first 60% of all the recorded spam associated with BGP agility behavior, most of the IPv6 prefixes are of a `/32` network. Ramachandran *et al.* originally observed and classified BGP spectrum agility in `/8` IPv4 networks, but no previous metric existed that suggested what we might expect to observe in our experiment regarding IPv6 network sizes. Using Table 4.5, we can strongly suggest that a `/32` would seem to be the most common type of prefix expected when observing BGP spectrum agility behavior in IPv6. Ramachandran *et al.* also found that the observed spamming IPv4 addresses were widely distributed across an advertised address space and that most the IPv4 addresses observed appeared only once. These discovered BGP agility characteristics by Ramachandran *et al.* are the antithesis of what we observed in the characteristics of our spammed IPv6 addresses. Most of the spam attempts within each prefix were sent from a small set of IPv6 addresses. This was surprising, as we expected to see a more significant exploitation of the large IPv6 address space by repeatedly changing source IP address. Moreover, multiple episodes of spamming attempts to the same destination IPv6 address can be observed over an extended time period.

Additionally, every IPv6 address type in our dataset has been allocated to a RIR and we did not observe the use of any "reserved" IPv6 prefixes. Finally, as demonstrated in Figure 4.5,

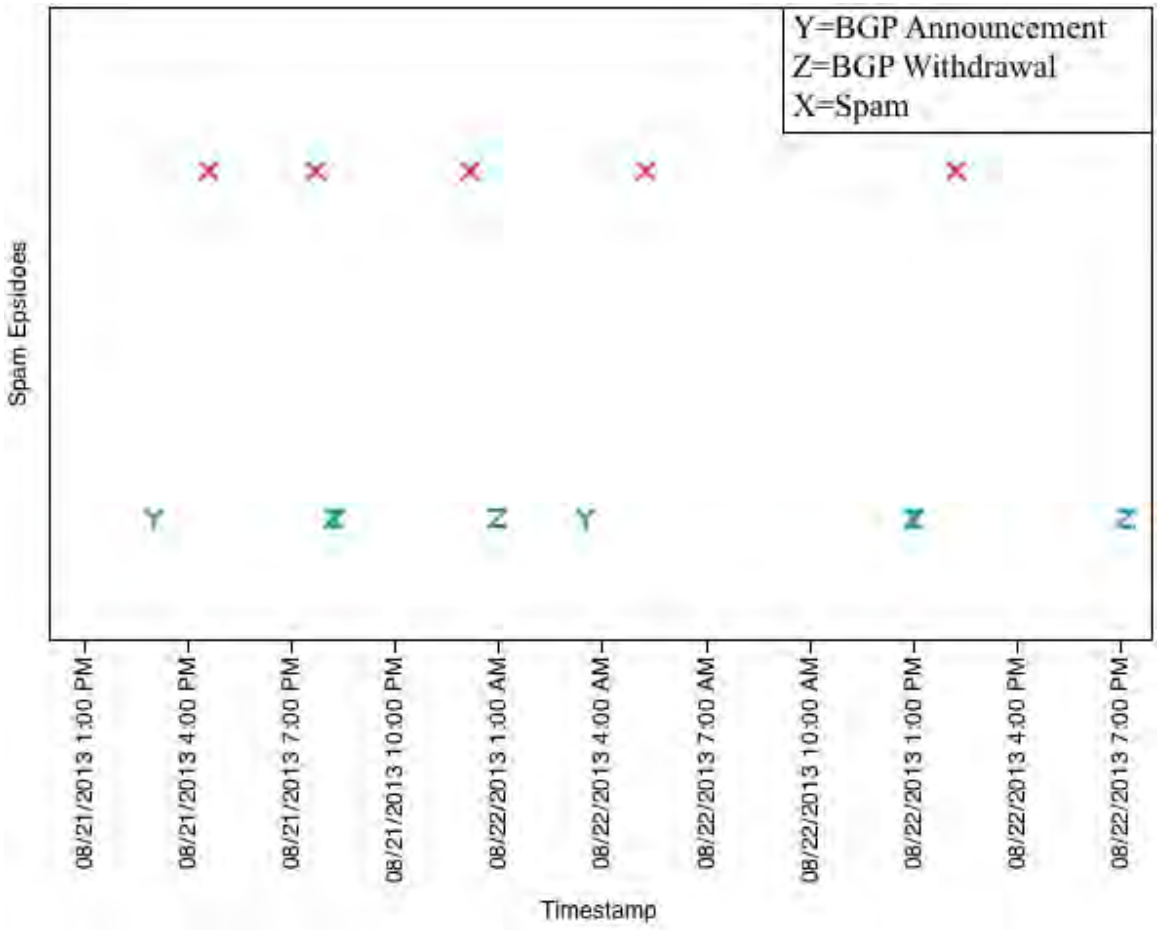

Figure 4.5: Observation of `2607:9000::/32` BGP Agility Behavior

we can see an instance in which a spammer attempts multiple spamming episodes using BGP spectrum agility in a short period of time.

While each spamming episode observed in our dataset is indicative of BGP spectrum agility, the episodes are not all uniform in their behavior. We viewed spamming episodes that attempted different IPv6 addresses within the prefix, some that only targeted one or two IPv6 addresses, and some that targeted IPv6 addresses that appear questionable in nature due to an address that uses "leetspeak" formatting. While the observed behaviors are certainly interesting and warrant study, the results are inconclusive as to whether or not all of our observed instances are IPv6 spammers intent on evading filtration mechanisms. We fully expected to catch IPv6 spammers that took advantage of the almost unlimited IPv6

address space by using different IPv6 addresses for every spam attempt within an episode, but we observed the opposite behavior; what appeared to be multiple instances of a target set of one to four IPv6 spamming addresses in an episode that differed by prefix. It appears that spammers are attempting to use the same methods that allowed them to spam in IPv4 in IPv6. In order to fully understand and classify this behavior a more comprehensive study needs to be conducted over an extended period of time to determine a more definitive answer to the IPv6 BGP spectrum agility question.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Conclusion

This thesis sought to investigate abusive IPv6 traffic by analyzing suspicious activity at the MTA of a production email domain and associating it with BGP routing updates. Using two experimental designs, one from a live production domain (referred to as `example.com`) and one from a laboratory (referred to as `test.com`), we were able to collect a corpus of IPv6 spammers and their behaviors. Additionally, we developed an IPv6 spam BGP correlation algorithm that utilized collected spam attempts to identify BGP spectrum agility behavior previously discovered as a method for malicious users to send abusive IPv4 traffic. We performed statistical analysis on our `example.com` dataset to recognize spamming profiles in both IPv4 and IPv6, to attempt to associate IPv4 and IPv6 spamming addresses, and to better understand what behaviors an IPv6 spammer might present when attempting to send malicious traffic over an extended time period. We provided some insight into the security studies of IPv6 as it emerges as a more dominant IP within the Internet.

We found that although IPv6 was not nearly as prevalent as we expected, we determined that no one type of OS, IPv6 address space, or network origin dominated our IPv6 observations. We also observed that there does not seem to be any preference over which type of IPv6 access methodology is used, be it native or tunneling technologies. We also found that IPv6 abusive traffic was minimal within our dataset and therefore so were the exploits of the large IPv6 address space. In contrast, our results of the BGP correlation algorithm proved insightful. We were able to discover many short-lived BGP agility behaviors that certainly warrant further investigation and classification. Our thesis efforts will bolster the study of IPv6 abusive traffic measurement and it is our hope that our work will provide a solid foundation for future study and mitigation efforts against abusive IPv6 Internet practices.

5.1 Future Work

To the best of our knowledge, this thesis work is the first time that spam measurement in IPv6 has been conducted in a live production domain from the point-of-view of the MTA. Additionally, this is the first publicly available study that seeks to correlate abusive IPv6 traffic with nefarious BGP behaviors. While our methods for measurement and study were

unique in nature and necessary as IPv6 takes a more active role in the Internet, there are some improvements that could be made to better characterize abusive IPv6 traffic:

Collecting the content of the spam messages from `example.com`. One of the shortcomings from the `example.com` measurement is the IPv4 and IPv6 spamming address association and any false positives or negatives that may have been contained within measurement constraints. One way to correctly classify whether or not traffic is spam is to collect and analyze message content. If this were to be done, we could not only ensure that legitimate mail is not being contained, but a matching algorithm could be used to better associate IPv4 and IPv6 spamming addresses assuming that a spammer would send the same spam content in IPv4 and IPv6. Finally, analyzing message content would allow us to better refine our filtration methods, which in turn would improve the accuracy of our data.

Further investigation into IPv6 spamming ASNs. A decent amount of follow up work could be conducted in the association of IPv6 spamming ASNs. Using the data from our BGP correlation algorithm, we could investigate trends over time to determine if there are certain dominant IPv6 spamming ASNs. Further investigation could determine if IPv6 spamming ASNs are consistent with IPv4 spamming ASNs and to determine if the rollout of IPv6 availability has any relation to IPv6 spamming entities.

Greater period of data collection and measurement analysis. Unfortunately, there was a limited amount of spam observed from `example.com` and the trends observed from the BGP correlation algorithm were not quite as dynamic as expected. As a result, more of these measurements need to be conducted over longer periods of time. As IPv6 expands, more data could be collected, allowing us a better chance of discovery of IPv6 spamming behaviors. Prevalence in certain areas of the Internet, measured growth over time, and new IPv6 spamming methods are all areas of measurement that will assist in identifying, characterizing, and monitoring spam in IPv6.

Appendix

```
#!/usr/bin/env python
#/*********************************************************************
#   Program:  $Id: playerv3.py 258 2014-08-11 20:16:39Z rbeverly $
#   Authors:  Robert Beverly <rbeverly@nps.edu> / Mark Turner <mjturner@nps.edu>
#   Purpose:  Determine lifetime of the IPv6 BGP prefixes announcing
#             observed IPv6 spam
# Pseudo-code:
#
# tree = new(radix trie)
# last_spam_time = 0
# for spam in sorted(spams):
#   current_spam_time = time(spam)
#   playAllBGP(last_spam_time, current_spam_time)
#   addspam(spam, tree)
#*********************************************************************
import sys
import os
import subprocess
import radix
import csv
import time
import datetime

class BGPUpdates:
  def __init__(self, dumpdir, verbose=False):
    self.bgpdump="/home/libbgpdump-1.4.99.13/bgpdump"
    self.dumpdir=dumpdir
    self.verbose=verbose
    self.rtree = radix.Radix()
    self.files = [ f for f in os.listdir(self.dumpdir) if f.find('updates.') != -1]
    self.files.sort()
    self.last_update_time = 0
    self.current_file = None
    self.proc = None
    self.grabNextFile()
    self.spamdict = dict()

  def processUpdates(self, end):
    print "> Processing BGP Updates from:", self.last_update_time, "to:", end
    (announce, withdraw) = (0,0)
    while (self.last_update_time < end):
      (tstamp, msgtype, prefix, origin) = self.grabOneUpdate()
      if self.verbose: print "tstamp:", tstamp, "msgtype:", msgtype,
      "prefix:", prefix, "origin:", origin
      self.updateTrie(tstamp, msgtype, prefix, origin)
      if msgtype == 'A': announce+=1
```

```
    elif msgtype == 'W': withdraw+=1
    self.last_update_time = tstamp
  return (announce, withdraw)

def updateTrie(self, tstamp, msgtype, prefix, origin):
  if msgtype == 'A':
    if not self.rtree.search_exact(prefix):
      rnode = self.rtree.add(prefix)
      rnode.data["spams"] = []
      rnode.data["announced"] = tstamp
      rnode.data["origin"] = origin
    #else:
    #  print "Announcement for existing prefix", prefix
  elif msgtype == 'W':
    rnode = self.rtree.search_exact(prefix)
    if rnode:
      spams = rnode.data["spams"]
      announced = rnode.data["announced"]
      life = tstamp - announced
      #if (life > 1000):
      #  print "\tWithdraw:", prefix, "announced:", announced, "lifetime:", life
      if (len(spams) > 0 && life <=90000) or self.verbose:
        print "\tWithdraw:", prefix, "announced:", announced,
        "withdrawn:", tstamp, "lifetime:", life
        print "\tWith associated spam messages:"
        for spamid in spams:
          print "\t\t", self.spamdict[spamid]
      self.rtree.delete(prefix)
  else:
    print "Unknown BGP message type."
    sys.exit(0)

def grabNextFile(self):
  if len(self.files) <= 0:
    if self.verbose: print "Reached end of update files."
    return False
  nextfile = self.files.pop(0)
  self.current_file = self.dumpdir + '/' + nextfile
  if self.verbose: print "Opening:", self.current_file
  self.proc = subprocess.Popen([self.bgpdump, "-m", self.current_file],
  stdout=subprocess.PIPE, stderr=subprocess.PIPE)
  return True

def grabOneUpdate(self):
  (tstamp, msgtype, prefix, originAS) = (None, None, None, None)
  while (True):
    line = self.proc.stdout.readline().strip()
    if len(line) == 0:
      if self.grabNextFile() == False:
```

```
            break
          line = self.proc.stdout.readline().strip()
        # for some reason, there are corrupt records in the MRT
        pureascii = ''.join([i if ord(i) < 128 else '' for i in line])
        if len(line) != len(pureascii):
          print "Line was corrupt:", line
          continue
        # at this point, we have ascii.  check number of tokens.
        tokens = line.split('|')
        if (len(tokens) != 15) and (len(tokens) != 6):
          print "Bad number of tokens:", len(tokens), "Line:", line
          continue
        (tstamp, msgtype, prefix) = (tokens[1], tokens[2], tokens[5])
        tstamp = int(tstamp)
        originAS = 0
        # invariant: at this point, we have a good MRT record.
        if msgtype=='A':
          try:
            aspathstr = tokens[6].split()
            origin = aspathstr[-1].translate(None, '{}')
            # deal with AS sets
            if origin.find(',') != -1:
              as_set = origin.split(',')
              origin = as_set[-1]
            originAS = int(origin)
          except Exception, e:
            print "Error:", e, "Line:", line
            continue
        # all good at this point.  break out of while and return
        break
      return (tstamp, msgtype, prefix, originAS)

  def addSpam(self, tstamp, addr, spamid):
    rnode = self.rtree.search_best(addr)
    if rnode:
      rnode.data["spams"].append(spamid)
      print "* Adding SpamID:", spamid, "to prefix:", rnode.prefix
      self.spamdict[spamid] = (tstamp, addr)
    else:
      print "- Couldn't find route to:", addr

if __name__ == "__main__":
  abusive6='/home/research/abusive6/'
  bgp = BGPUpdates(dumpdir=abusive6+'bz2dump', verbose=True)
  csvfile = open('blah.csv', 'rU')
  spamreader=csv.reader(csvfile, delimiter=',')
  spamid = 0
  for row in spamreader:
    (timestring, addr) = row
```

```
    tstamp = int(time.mktime(time.strptime(timestring, "%m/%d/%y %I:%M %p")))
    print "> New spam. Timestamp:", tstamp, "Addr:", addr
    (a,w) = bgp.processUpdates(tstamp)
    bgp.addSpam(tstamp, addr, spamid)
    spamid+=1
    print "* Processed", a, "BGP announcements,", w, "withdrawals."
csvfile.close()
```

List of References

[1] Internet Corporation for Assigned Names and Numbers. (2011). Available pool of unallocated ipv4 internet addresses now completely emptied. [Online]. Available: www.icann.org/en/news/releases/release-03feb11-en.pdf

[2] G. Huston. (2013, February). Ipv4 address report. [Online]. Available: http://www.potaroo.net/tools/ipv4/index.html

[3] Google. (2014). Google ipv6 adoption. [Online]. Available: http://www.google.com/ipv6/statistics.html#tab=ipv6-adoption

[4] P. Srisuresh and K. Egevang, "Traditional ip network address translator (traditional nat)," Internet Requests for Comments, RFC Editor, RFC 3022, January 2001. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3022.txt

[5] S. Deering and R. Hinden, "Internet protocol, version 6 (ipv6) specification," Internet Requests for Comments, RFC Editor, RFC 2460, December 1998. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2460.txt

[6] "Planning guide/roadmap toward ipv6 adoption within the u.s. government," The Federal CIO Council Strategy and Planning Committee, July 2012. [Online]. Available: http://www.ipv6forum.com/dl/presentations/USGv6Roadmap.pdf

[7] RIPE-NCC. (2012). IPv6 enabled networks. [Online]. Available: http://v6asns.ripe.net/v/6

[8] M. V. Hauser, "Attacking the ipv6 protocol suite," *The Hacker's Choice*, 2006.

[9] K. Barker, "The security implications of IPv6," *Network Security*, vol. 2013, no. 6, pp. 5–9, 2013.

[10] A. Afanasyev, N. Tilley, B. Longstaff, and L. Zhang, "BGP routing table: Trends and challenges," in *Proceedings of the 12th Youth Technological Conference High Technologies and Intellectual Systems*, Moscow, Russia, April 2010. [Online]. Available: http://www.iu4.ru/konf/2010_ts/01_program.pdf

[11] P. Kosik, P. Ostrihon, and R. Rajabiun, "Ipv6 and spam," in *Proceedings of the 2009 MIT Spam Conference*, 2009.

[12] K. Claffy, "Tracking IPv6 evolution: Data we have and data we need," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 3, pp. 43–48, 2011. [Online]. Available: http://doi.acm.org/10.1145/2002250.2002258

[13] A. Dhamdhere, M. Luckie, B. Huffaker, A. Elmokashfi, E. Aben *et al.*, "Measuring the deployment of ipv6: topology, routing and performance," in *Proceedings of the 2012 ACM conference on Internet measurement conference*. ACM, 2012, pp. 537–550.

[14] ISOC. (2012). World ipv6 launch. [Online]. Available: http://www.worldipv6launch.org

[15] L. F. Cranor and B. A. LaMacchia, "Spam!" *Communications of the ACM*, vol. 41, no. 8, pp. 74–83, Aug. 1998. [Online]. Available: http://doi.acm.org/10.1145/280324.280336

[16] (2011). Spam: Unsolicited email messages. [Online]. Available: http://www.pandasecurity.com/homeusers/security-info/types-malware/spam/

[17] (2014). 6to4. The IPv6 Portal by Consulintel. [Online]. Available: http://www.ipv6tf.org/index.php?page=using/connectivity/6to4

[18] K. Lougheed and Y. Rekhter, "A border gateway protocol (bgp)," Internet Requests for Comments, RFC Editor, RFC 1105, June 1989. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1105.txt

[19] D. Meyer, "Route views," University of Oregon. Advanced Network Technology Center, 2014. [Online]. Available: http://routeviews.org

[20] A. Ramachandran and N. Feamster, "Understanding the network-level behavior of spammers," in *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4. ACM, 2006, pp. 291–302.

[21] S. Zander, L. L. Andrew, G. Armitage, G. Huston, and G. Michaelson, "Mitigating sampling error when measuring internet client ipv6 capabilities," in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, ser. IMC '12. New York, NY, USA: ACM, 2012, pp. 87–100. [Online]. Available: http://doi.acm.org/10.1145/2398776.2398787

[22] H. Rafiee, M. von Löwis, and C. Meinel, "Ipv6 deployment and spam challenges," *IEEE, Internet Computing*, vol. 16, no. 6, pp. 22–29, Nov 2012.

[23] J. Klensin, "Simple mail transfer protocol," Internet Requests for Comments, RFC Editor, RFC 5321, October 2008. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5321.txt

[24] R. Thomas. (2014). Team cymru bogon route-server project. [Online]. Available: http://www.cymru.com/

[25] B. Carpenter, "Advisory guidelines for 6to4 deployment," RFC Editor, RFC 6343, August 2011. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6343.txt

[26] M. Zalewski. (2012). Passive OS fingerprinting tool. [Online]. Available: http://lcamtuf.coredump.cx/p0f3/

[27] M. Nakamura and J. Hagino, "Smtp operational experience in mixed ipv4/v6 environments," Internet Requests for Comments, RFC Editor, RFC 3974, January 2005. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3974.txt

[28] R. Draves, "Default address selection for internet protocol version 6 (ipv6)," Internet Requests for Comments, RFC Editor, RFC 3484, February 2003. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3484.txt

[29] D. Thaler, R. Draves, A. Matsumoto, and T. Chown, "Default address selection for internet protocol version 6 (ipv6)," Internet Requests for Comments, RFC Editor, RFC 6724, September 2012. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6724.txt

[30] M. Corporation. (2014). Microsoft exchange server 2010. [Online]. Available: http://www.microsoft.com/en-us/download/details.aspx?id=21308

[31] I. Sendmail. (2014). Sendmail open source. [Online]. Available: http://www.sendmail.com/sm/open_source/

[32] W. Venema. (2014). Postfix. [Online]. Available: http://www.postfix.org/

[33] M. Cotton, L. Eggert, J. Touch, M. Westerlund, and S. Chesire, "Internet assigned numbers authority (iana) procedures for the management of the service name and transport protocol port number registry," Internet Requests for Comments, RFC Editor, RFC 6335, August 2011. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6335.txt

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia
2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California